

REMARKS/ARGUMENTS

Claims 1, 3-5, 16, 18, 21, 28, 30, 35, 37, 41, 43, 46, 47, 50 and 53 are amended. Claims 2, 11-15, 27, 42, 48 and 51-52 are canceled without prejudice. Therefore claims 1, 3-10, 16-26, 28-41, 43-47, 49-50, and 53 are presented for examination. Applicants request reconsideration of the current application in light of the above amendments and the following remarks.

Objections to the Specification

The November 20, 2006 non-final office action (“Office Action”) objects to the use of the term “Java” in the specification. (Office Action, p. 2). The Office finds that “Java” is a trademark that should be designated as “JavaTM.” Applicants located the term “Java” only in Paragraph 9 on page 8 of the specification. That paragraph is amended to refer to “JavaTM.” The objection is thus obviated and Applicants respectfully request that the objection to the specification be withdrawn.

Claim Rejections under 35 U.S.C. § 101

Claims 32-40

The Office Action rejects claims 32-40 under Section 101 arguing that independent claim 32 “raises a question” of whether it is directed to an “abstract idea.” (Office Action, p. 3, ¶ 7).

Claim 32 recites:

32. An apparatus comprising:
 - a front-end code generator to transform source code into intermediate code and provide the intermediate code to a profiler; and
 - the profiler, coupled with the front-end code generator, to receive external execution input, execute the intermediate code using the external execution input,

generate a performance profile regarding the performance of the intermediate code, and annotate the intermediate code based, at least in part, on the performance profile, to generate annotated intermediate code;

a back-end code generator, coupled with the profiler, to receive the annotated intermediate code, and transform the annotated intermediate code into machine code.

Thus, claim 32 recites a front end to provide intermediate code to a profiler, a profiler to generate annotated intermediate code from the intermediate code, and a back-end generator to transform the annotated intermediate code into machine code. As explained in the specification (*See, e.g.*, paragraphs 25-26), this machine code executes faster than machine code that is conventionally produced.

Thus, claim 32 recites elements that produce faster executing (e.g., optimized) machine code – a useful, concrete, and tangible result. Claim 32 therefore recites statutory subject matter. *Interim Guidelines for Examination of Patent Applications for Patent Subject Matter Eligibility* (“Interim Guidelines”), p. 1, citing *State Street Band & Trust Co. v. Signature Financial Group Inc.*, 149 F.3d 1368, 1373-74 (Fed. Cir. 1998).

Faster executing code increases computer efficiency. The *Interim Guidelines* recognize that a claim to a data structure on a computer readable medium that “**increases computer efficiency**” is statutory. *Interim Guidelines*, Annex IV, p. 50, citing *In re Lowery*, 32 F.3d 1579, 1583-84, 32 U.S.P.Q.2d 1031, 1035 (Fed. Cir. 1035) (Emphasis added).

Therefore, claim 32, which recites elements that produce faster executing machine code is statutory.

The Office Action argues that because claim 32 recites an apparatus without reciting a processor or memory, it is claiming “software per se” that is not tangibly embodied and amounts only to an abstract idea. However, nothing requires that a claim recite a tangible physical article or machine to be patentable subject matter. While the *Interim Guidelines* limit the patentability of “computer programs claimed as computer listings per se” (*Interim Guidelines* p. 53, emphasis in original), that rule is rather limited. By its own terms, it applies only to computer programs claimed as computer listings per se. Clearly, computer listings per se by their very nature cannot produce a useful, concrete, or tangible result. Therefore, if a claimed element produces a useful, concrete and tangible result, it is not a computer listing per se.

Claim 32 does not claim computer listings per se. Claim 32 recites elements that ultimately produce faster executing machine code. Thus, the claimed elements are capable of producing a useful, concrete and tangible result. They are not mere “computer listings per se.”

The focus should not be on whether claim 32 recites specific hardware. Rather, the determinative issue is whether the recited elements produce a practical result – which they do. “Thus, the question of whether a claim encompasses statutory subject matter should not focus on which category of subject matter a claim is directed (e.g., process or machine), ‘but rather on the essential characteristics of the subject matter, **in particular its practical utility.**” *Interim Guidelines*, p. 37, quoting the *State Street* decision cited above.

Again, because the recited elements of claim 32 have practical utility – producing faster machine code – the claim recites statutory subject matter. Dependent claims 33-40

are deemed to recite the same statutory subject matter. Applicants respectfully request that the Section 101 rejection of claims 32-40 be withdrawn.

Claims 41-49

The Office rejects claims 41-49 because they recite a machine accessible medium and because a “machine readable medium” is defined at paragraph [0014] of the specification to include “intangible media.” (Office Action, p. 3). The rejection of claims 42 and 48 is moot because they have been canceled.

As for the remaining claims, 41, 43-47, and 49, Applicant has amended independent claim 41 to recite a “computer readable medium” instead of a “machine accessible medium.” A computer readable medium is recognized as patentable subject matter. Dependent claims 43-47, and 49 are deemed to recite the same limitation. The rejections are thus obviated. Applicant respectfully requests that the rejection of claims 41, 43-47, and 49 be withdrawn.

Claim Rejections under 35 U.S.C. § 102(e)

The Office Action rejects claims 1-22, 24-26, 32-46, and 50-53 under Section 102(e) as being anticipated by U.S. Patent No. 6,289,505 to Goebel (“Goebel”). The rejection of claims 2, 11-15, 42, and 51-52 is moot because these claims have been canceled without prejudice. Applicants assert that the remaining rejected claims 1, 3-10, 16-22, 24-26, 32-41, 43-46, 50 and 53 are not anticipated by Goebel for at least the reasons stated below.

“A claim is anticipated only **if each and every element as set forth in the claim** is found, either expressly or inherently described, in a single prior art reference.” MPEP § 2131. Further, “[t]he **identical invention** must be shown in **as complete detail as is contained in the ... claim.**” *Id.* (emphasis added).

Claim 1, as amended, recites:

1. A method comprising:
receiving source code;
transforming the source code to intermediate code;
executing the intermediate code;
**generating data that indicates performance of the executed
intermediate code; and**
producing machine code based on the data and the intermediate code.
(Emphasis added).

Thus, the emphasized portions of claim 1 recite generating data (hereinafter, “performance data”) that indicates the performance of executed intermediate code and then producing machine code based on the performance data and the intermediate code. Goebel does not teach at least these limitations of claim 1 and therefore does not anticipate claim 1.

The generated performance data recited in claim 1 indicates the performance of executed intermediate code – not binary executable code. Machine code is then produced based on the performance data and the intermediate code. In contrast, Goebel describes processing profile information generated during the execution of binary executable code.

For example, at col. 6, lines 61-66, Goebel states that, “The intermediate representation optimizer segment 305 of the re-optimizing compiler 300 can also process profile information 317 **generated during execution of an instrumented binary executable** to determine which portions of the binary executable most need to be optimized” (emphasis added). This portion of Goebel describes using profile information generated during the execution of a binary executable. It does not describe producing machine code from performance data that indicates the performance of executed intermediate code, as recited by claim 1. Applicants are unaware of any portion of Goebel that describes the above limitations of claim 1.

In rejecting the previous unamended version of claim 1, the Office cites Fig. 3, element 307 (and related discussion) of Goebel as describing “generating data that indicates the performance of executed code.” (Office Action, p. 4, ¶ 9). However, as amended, claim 1 recites generating data that indicates the performance of “executed **intermediate** code” – not just of executed code. Turning to the cited Fig. 3, element 307, it merely describes “Code Generator and Low Level Optimizer.” It does not even describe performance data, much less generating performance data that indicates the performance of executed intermediate code, as recited by amended claim 1.

Although not cited by the Office, Element 317 of Fig. 3 of Goebel describes “Profile Information” with an arrow pointing to Element 305, describing an “Intermediate Representation Optimizer.” However, neither Element 317 nor Element 305 describe generating data that indicates the performance of executed intermediate code, as recited by amended claim 1.

The Office also cites the “related discussion” of Fig. 3 in support of its rejection. Fig. 3 is described at col. 6, line 22 – col. 7, line 2 of Goebel, which provides a general discussion of a purported re-optimizing compiler 300. However, this cited portion of Goebel has no discussion of generating performance data that indicates the performance of executed intermediate code. The only part of this cited passage of Goebel that discusses profile information is at col. 6, lines 61-66, which is discussed above as not describing generating performance data that indicates the performance of executed intermediate code. Thus, the cited related discussion of Fig 3 also does not describe the above limitations of amended claim 1.

In rejecting the unamended version of claim 1, the Office also cites Fig.3, Elements 305-309 (and related discussion) as describing “causing the executed code to be

modified based, at least in part, on the data.” (Office Action, p. 4, ¶ 9). In the amended version of claim 1, the above-quoted claim language has been replaced with new claim language that recites “producing machine code based on the data and the intermediate code.”

Reviewing Fig. 3, and in particular, Elements 305-309, there is no description of the above limitation of amended claim 1. Element 305 refers to “Intermediate Representation Optimizer”, Element 307 to “Code Generator and Low Level Optimizer” and Element 309 to “Binary Module.” Neither these elements nor the rest of Fig. 3 refer to producing machine code based on performance data and intermediate code, where the performance data indicates the performance of executed intermediate code, as recited in amended claim 1.

The text related to Fig 3 has already been discussed as not describing the generation of performance data that indicates the performance of intermediate code. It certainly does not describe using such performance data, and the intermediate code, to generate machine code. Instead, as discussed above, it describes generating profile information during execution of an instrumented binary executable.

Thus, neither Fig. 3, Elements 305-309, nor the related discussion of Fig.3 describe at least the above limitations of claim 1.

In rejecting canceled dependent claim 2 – which recited executing and modifying intermediate code – the Office cites Fig. 3, element 305 of Goebel. (Office Action, p. 4, regarding claim 2). However, Fig. 3, element 305 is already discussed above as not describing performance data that indicates the performance of executed intermediate code.

Therefore, Goebel does not teach, either expressly or inherently, at least the above limitations of amended claim 1. Goebel thus does not anticipate claim 1.

The above argument regarding claim 1 also applies to independent claims 18, 32, 41 and 50. Claim 18 similarly recites “providing the intermediate code to a profiler that executes the intermediate code and generates annotated intermediate code based on the performance of executed intermediate code.” Claim 18 further recites transforming the annotated intermediate code into machine code. Independent claims 32, 41, and 50 recite performance data, or a performance profile that indicates the performance of intermediate code. Thus, the above discussion is thus fully applicable to those claims. Independent claims 18, 32, 41 and 50 are therefore also not anticipated by Goebel.

Independent claims 21, 35 and 46 are distinguishable from Goebel on a different basis.

Independent claim 21, as amended, recites:

21. A method comprising:
producing machine code based upon source code;
receiving a data file generated by a profiler, wherein the data file indicates a performance of the machine code as executed by the profiler;
producing modified machine code based on the source code and the data file; and

iteratively:

determining whether to produce further modified machine code; and, if further modified machine code is to be produced:

providing the modified machine code to the profiler;
receiving another data file from the profiler; and
producing further modified machine code based upon the source code and the another data file.

(Emphasis added).

Thus, the emphasized portions of claim 21 recite iteratively determining whether to produce further modified machine code. And if the further modified machine code is to be produced, then providing the modified machine code to the profiler, receiving another data file from the profiler, and producing further modified machine code based upon the source code and the another data file. Goebel does not describe at least the above limitations of claim 21 and therefore does not anticipate claim 21.

The Office Action admits that Goebel does not teach determining whether to further modify the modified machine code. Although this limitation was not recited in the previous unamended version of claim 21, it was recited in now-canceled dependent claim 27. In rejecting now-canceled dependent claim 27 for obviousness under Section 103(a), the Office admits, “Goebel does not explicitly disclose determining whether to modify the modified machine code; and providing the modified machine code to the profiler, if the modified machine code is to be further modified.” (Office Action, p. 15, regarding claim 27).

Although claim 27 has been canceled, its limitations, with some modification, are now recited in amended independent claim 21. Unlike dependent claim 27, amended claim 21 claims performing the determining “iteratively.” Amended claim 21 additionally recites that if further modified machine code is to be produced, then producing the further modified machine code, including providing the modified machine code to the profiler.

The Office argues that the above-discussed limitation of now-canceled claim 27 is described by U.S. Patent No. 6,874,410 to Shupak (“Shupak”). (Office Action, pp. 14-15, regarding claim 27). However, a review of the cited portions of Shupak indicates that it fails to describe the above limitations as recited in amended claim 21.

The Office cites Fig. 7 and the related discussion of Fig. 7 as describing the above limitation. (Office Action, p. 15, regarding claim 27). Fig. 7 of Shupak is a flow chart with element 710 that describes reading annotation information in an executable computer program and element 720 that describes modifying the executable program in accordance with the annotation information. While it could be argued that Fig. 7 describes reading annotation information to learn how to modify an executable program, there is nothing in Fig. 7 that describes determining whether to modify machine code. Fig. 7 also does not describe the executable program as modified machine code. That is, there is nothing in Fig. 7 that states that the executable program already has been modified.

Further, Fig. 7 does not describe that if the program is to be further modified, then providing the program to a profiler. Instead, under Fig. 7, the executable computer program is just modified in accordance with the annotation information.

Fig. 7 also does not describe receiving another data file from the profiler and producing further modified machine code based on the data file and the source code.

Nor is there anything in Fig. 7 that describes performing any determining iteratively. Thus, Fig. 7 of Shupak fails to describe at least each of the above limitations of amended claim 21.

Turning to the col. 10, line 60 to col. 11 line 45 of Shupak – which describes Fig. 7 – there is also nothing the describes iteratively determining whether to further modify already modified machine code. The passage describes possible sources for the annotation information. It further describes “modifying the executable program to perform an action in accordance with the information in the annotation debug information.” (Shupak, col. 11, lines 4-5).

But this passage also lacks the specifics discussed above. These include iteratively determining whether to produce further modified machine code, providing modified machine code to a profiler, and producing further modified machine code based on the data file received from the profiler and the source code.

Thus, Fig. 7 and the related text fail to describe at least the above limitations of amended claim 21. Applicants are unaware of any other portion of Shupak that describes these limitations. Therefore, the cited portions of Goebel and Shupak fail to teach or suggest at least the above limitations of claim 21. Thus, claim 21 is not rendered obvious by the combination of Goebel and Shupak.

Further, the Office would fail to make a *prima facie* case of obviousness by combining these two references. In rejecting now-canceled dependent claim 27 under Section 103 over Goebel in view of Shupak, the Office fails to describe a sufficient motivation or suggestion for combining the two references. The Office cites col. 3, lines 25-40 as describing a motivation to modify modified machine code to avoid the overhead execution time. (Office Action, p. 15). However, this cited portion of Shupak does not describe modified machine code. Also, the overhead execution time that is discussed relates to that created by including extra annotation code into object code – such as extra “C runtime printf calls.” (Shupak, col. 3, lines 25-28). This cited passage fails to provide any suggestion or motivation for combining the two references.

Thus, even if the above limitations were taught or suggested by the combined references, there is no demonstrated motivation or suggesting for combining the two references. This is a further reason why claim 21 is not rendered obvious by the combination of the two references.

Independent claims 35 and 46 similarly recite determining whether to further modify modified machine code, and if so, to provide modified machine code to a profiler, to receive another data file from the profiler and to produce further modified machine code. The above discussion regarding claim 21 is applicable to these independent claims. For the above reasons, independent claims 35 and 46 are also not rendered obvious by the combination of Goebel and Shupak.

Dependent claims 3-10, 16-17, 19-20, 22, 24-26, 33-34, 36-40, 43-45, and 53 depend from one of independent claims 1, 18, 21, 32, 35, 41, 46 and 50. They are deemed to recite the limitations of their respective independent claims. Dependent claims 3-10, 16-17, 19-20, 22, 24-26, 33-34, 36-40, 43-45, and 53 are thus also not rendered obvious by the combination of Goebel and Shupak.

Rejections under 35 U.S.C. § 103(a)

Claims 23 and 24

The Office Action rejects claims 23 and 24 under Section 103(a) as being unpatentable over Goebel and alleged prior art admissions in the specification. (Office Action, p. 13, ¶ 11). However, claims 23 and 24 depend from independent claim 21. As discussed above, Goebel does not teach or suggest all of the limitations of independent claim 21. The alleged prior art admissions in the specification are not cited as teaching or suggesting the limitations of independent claim 21. Therefore, claim 21 is not rendered obvious by the combination of Goebel and the allegedly admitted prior art. Thus, dependent claims 23 and 24 are also not rendered obvious by the combination of Goebel and the allegedly admitted prior art. MPEP § 2143.03.

Claims 27-31 and 47-49

The Office Action rejects dependent claims 27-31 and 47-49 under Section 103(a) as being unpatentable over Goebel in view of Shupak. (Office Action, p. 14, ¶ 12). The rejection of claims 27 and 48 is moot because those claims are canceled without prejudice.

However, the remaining dependent claims 28-31, 47 and 49 depend from one of independent claims 21 and 46. As discussed above, Goebel does not teach or suggest all of the limitations of independent claim 21 and 46. Further, Shupak is not cited as teaching or suggesting the limitations of independent claims 21 and 46.

As discussed above, the limitation of canceled dependent claim 27 is now recited, with modification, in amended claim 21. Those limitations are also recited, with modification, in amended claim 46. However, as discussed above, those limitations are neither taught nor suggested by the combination of Goebel and Shupak. Thus, independent claims 21 and 46 are not rendered obvious by the combination of Goebel and Shupak. Dependent claims 28-31, 47 and 49 are deemed to recite the limitations of independent claims 21 and 46. Claims 28-31, 47 and 49 are therefore not rendered obvious by the combination of Goebel and Shupak.

CONCLUSION

In view of the foregoing, it is respectfully asserted that all of the claims pending in this patent application are in condition for allowance.

Should it be determined that an additional fee is due under 37 CFR §§1.16 or 1.17, or any excess fee has been received, please charge that fee or credit the amount of overcharge to deposit account #02-2666.

If the Examiner has any questions, he is invited to contact the undersigned at (503) 439-8778. Reconsideration of this patent application and early allowance of all the claims is respectfully requested.

Respectfully submitted,

Date: Feb. 15, 2007

/Jose R. Mata/
Jose R. Mata
Reg. No. 56,978

Blakely, Sokoloff, Taylor & Zafman, LLP
12400 Wilshire Boulevard
Seventh Floor
Los Angeles, CA 90025-1026